mL

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,361	10/22/2003	Peter-Pike J. Sloan	3382-66857	9370

26119 7590 03/08/2007

KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER	
BROOME, SAID A	
ART UNIT	**PAPER NUMBER**
2628	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/08/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/692,361	SLOAN ET AL.	
	Examiner	Art Unit	
	Said Broome	2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 November 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to an amendment filed 11/30/2006.

2. Claims 1-20 are original.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 3, 4, 6-8, 10-12 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan et al.(hereinafter "Sloan", "*Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments*") in view of Burke (US 2003/0063096) in further view of Purcell et al.(hereinafter "Purcell", "*Ray Tracing on Programmable Graphics Hardware*").

Regarding claim 1, Sloan teaches all the limitations except creating object positions and normal texture. Sloan teaches computing radiance transfer functions for each set of directions sampled about the object on page 1 second column second paragraph lines 1-8 (" …we have a convex, diffuse object lit by an infinitely distant environment map. The object's "response" to its environment can be viewed as a transfer function, mapping incoming and outgoing radiance…A more complex integral captures how a concave object shadows itself, where the integrand is

multiplied by an additional transport factor representing visibility along each direction."). Sloan also teaches rendering the object from the direction to produce a shadow buffer representing depth from the object in the direction for the set of points on page 2 first column third paragraph lines 1-3 ("Shadow maps, containing depths from the light source's point of view…") and in the caption of Figure 2 lines 4-6 ("…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…"). Sloan also teaches determining radiance transfer contribution of the set of sampled points for the currently iterated direction based on the determined cosine terms and shadowing on page 1 second column second paragraph lines 2-8 ("…object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral. A more complex integral captures how a concave object shadows itself, where the integrand is multiplied by an additional transport factor representing visibility along each direction."), where it described that the radiance transfer is computed for all directions in response to a integral comprising the cosine terms and shadowing effects. Sloan teaches accumulating the radiance transfer contributions of the set of sampled points for the currently iterated direction with that of previously iterated direction on page 5 second column second paragraph lines 1-7 ("For diffuse surfaces, at each point pεO we further compute the transfer vector by SH-projecting…SH-projection to compute the transfers is performed by numerical integration over the direction samples s_d, summing into an accumulated transfer…"), where it is described that for each point on the surface the radiance transfer is accumulated for all directions and therefore computes the radiance transfer for the current as well as any previous direction. Sloan illustrates a rendered image of an object in a lighting

environment based on accumulated radiance transfer contribution that is presented in the right of Figure 1. Again, Sloan fails to teach creating an object positions and normal texture. Burke teaches creating an object positions texture representing positions of a set of points sample over the object mapped into a texture space and object normals texture representing normals of the set of sampled points mapped into the texture space in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")"). However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the method illustrated in Figure 3 of the applicant's Specification, because reversal of the loops would optimize the radiance processing when executed on graphics hardware, as taught by Purcell in section 3.2 4th paragraph lines 1-5 – 10-14 ("…we present an optimization to minimize the total number of passes…There are various strategies for nesting these loops…The following is a more efficient algorithm…"), where the nested loops are reversed to enable efficient execution on a graphics processor, as shown in the pseudo-code in section 3.2 in the right column. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Burke and Purcell because this combination would provide a radiance transfer process that iterates over a set of directions in the outer loop and iterates over sample points in the inner loop, thereby efficiently computing realistic self-shadowing and lighting effects through optimization of the process for improved functionality on graphics hardware.

Regarding claim 3, Sloan teaches all the limitations except creating object positions and normal texture. Sloan teaches graphics hardware processing software that computes radiance transfer on page1 second column third paragraph lines 1-7 ("The resulting transfer functions are represented as a dense set of vectors or matrices over its surface...The graphics hardware can dynamically sample incident radiance at a number of points."), on page 8 first column fourth paragraph lines 8-10 ("Using graphics hardware, incident lighting can be sampled every frame and at multiple points...") and on page 7 section 8 second paragraph lines 4-7 and second column lines 1-3 ("Our current implementation precomputes the transfer matrix p M at each point...we perform the matrix transform from equation (9) in software at each point...The result is a volume texture containing coefficients of transferred radiance..."), therefore the programming code, or software, must be embodied on a computer readable media because it is executed to render images of an object, as described on page 8 first column second paragraph lines 1-3 ("...images in this paper (Figures 1 and 3-12) all of which were computed with the PC graphics hardware."). Sloan teaches programming code for performing the contents of this reference on page 1 first column first paragraph lines 1-9 ("...global transport simulator creates functions over the object's surface...At run-time, these transfer functions are applied...") and on page 7 section 8 second paragraph lines 4-7 and second column lines 1-3 ("Our current implementation precomputes the transfer matrix p M at each point...we perform the matrix transform from equation (9) in software at each point...The result is a volume texture containing coefficients of transferred radiance..."), therefore code or software is utilized to perform all the succeeding limitations. Sloan also teaches rendering the object from the direction to produce a shadow buffer representing depth from the object in the direction for the set of points on page 2

first column third paragraph lines 1-3 ("Shadow maps, containing depths from the light source's point of view…") and in the caption of Figure 2 lines 4-6 ("…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…"). Sloan also teaches determining radiance transfer contribution of the set of sampled points for the currently iterated direction based on the determined cosine terms and shadowing on page 1 second column second paragraph lines 2-8 ("…object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral. A more complex integral captures how a concave object shadows itself, where the integrand is multiplied by an additional transport factor representing visibility along each direction."), where it described that the radiance transfer is computed for all directions in response to a integral comprising the cosine terms and shadowing effects. Sloan teaches accumulating the radiance transfer contributions of the set of sampled points for the currently iterated direction with that of previously iterated direction on page 5 second column second paragraph lines 1-7 ("For diffuse surfaces, at each point pεO we further compute the transfer vector by SH-projecting…SH-projection to compute the transfers is performed by numerical integration over the direction samples s_d, summing into an accumulated transfer…"), where it is described that for each point on the surface the radiance transfer is accumulated for all directions and therefore computes the radiance transfer for the current as well as any previous direction. Sloan illustrates a rendered image of an object in a lighting environment based on accumulated radiance transfer contribution that is presented in the right of Figure 1. Again, Sloan fails to teach creating an object positions and normal texture. Burke teaches creating an object positions texture representing positions of

a set of points sample over the object mapped into a texture space and object normals texture representing normals of the set of sampled points mapped into the texture space in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")"). However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the method illustrated in Figure 3 of the applicant's Specification, because reversal of the loops would optimize the radiance processing when executed on graphics hardware, as taught by Purcell in section 3.2 4th paragraph lines 1-5 – 10-14 ("...we present an optimization to minimize the total number of passes...There are various strategies for nesting these loops...The following is a more efficient algorithm..."), where the nested loops are reversed to enable efficient execution on a graphics processor, as shown in the pseudo-code in section 3.2 in the right column. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Burke and Purcell because this combination would provide a radiance transfer process that iterates over a set of directions in the outer loop and iterates over sample points in the inner loop, thereby efficiently computing realistic self-shadowing and lighting effects through optimization of the process for improved functionality on graphics hardware.

Regarding claim 4, Sloan teaches determining cosine terms on page 1 second column second paragraph lines 2-8 ("...object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a

cosine-weighted integral."), where it described that the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing on page 2 first column third paragraph lines 1-3 ("Shadow maps, containing depths from the light source's point of view…") and in the caption of Figure 2 lines 4-6 ("…a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing…"), and determining and accumulating radiance transfer contributions over the sampled points in each direction in the caption of Figure 2 lines 1-10 ("A transfer vector at a particular point on the surface represents how the surface responds to incident light at that point…matrix transforms the lighting coefficients into the coefficients of a spherical function…The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point…"). However, though Sloan teaches calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the method illustrated in Figure 3 of the applicant's Specification, because reversal of the loops would optimize the radiance processing when executed on graphics hardware, as taught by Purcell in section 3.2 4th paragraph lines 1-5 – 10-14 ("…we present an optimization to minimize the total number of passes…There are various strategies for nesting these loops…The following is a more efficient algorithm…"), where the nested loops are reversed to enable efficient execution on a graphics processor, as shown in the pseudo-code in section 3.2 in the right column. The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 6, Sloan fails to teach the limitations. Burke teaches an object positions texture arrangement of data values representing the position of each of the sampled points mapped into the texture space in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 7, Sloan fails to teach the limitations. Burke teaches that the object positions texture I stored in an RGB component format in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color… (e.g., "RGB")"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 8, Sloan fails to teach the limitations. Burke teaches object normals texture contains an arrangement of data values representing the surface normal at each of the sampled points mapped into the texture space in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ")…and normal information (e.g., "IJK")"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 10, Sloan teaches determining cosine terms on page 1 second column second paragraph lines 2-8 ("…object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a

cosine-weighted integral."), where it described that the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing on page 2 first column third paragraph lines 1-3 ("Shadow maps, containing depths from the light source's point of view...") and in the caption of Figure 2 lines 4-6 ("...a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing..."), and determining and accumulating radiance transfer contributions over the sampled points in each direction in the caption of Figure 2 lines 1-10 ("A transfer vector at a particular point on the surface represents how the surface responds to incident light at that point...matrix transforms the lighting coefficients into the coefficients of a spherical function...The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point..."). Sloan also teahces performing radiance transfer contributions on a pixel shader, as described on page 7 first column fifth paragraph lines 4-7 and second column first paragraph lines 1-5("At run-time, we perform the matrix transform from equation (9) in software at each point in the volume and upload the result to the graphics hardware. The result is a volume texture containing coefficients of transferred radiance (L'_p) which is applied to R. Then in a pixel shader this transferred radiance is used to light the receiver."), executable on a programmable graphics processing unit, such as the programmable graphics processor described on page 6 first column section 6 third paragraph line 6 ("DirectX 8.1 pixel shaders"). Therefore one of ordinary skill would have been capable of also determining cosine terms and shadowing using the pixel shader because they both contribute to the rendering of the radiance transfer.

Regarding claim 11, Sloan teaches rendering the object from the direction comprises the object as an orthographic camera projection whose view direction is set to the current direction on page 2 first column first paragraph lines 14-16 ("...evaluated at the view-dependent reflection direction to produce the final shading."), where it is described that the shading is perform based on the view therefore the object is rendered based on the particular view direction, as described on page 6 first column section 6 first paragraph line 3 and step 4 ("Rendering O requires the following steps at run-time:... the radiance vector resulting from step 3 is convolved with O's BRDF at p, and then evaluated at the view-dependent reflection direction R").

Regarding claim 12, Sloan teaches that the occlusion and shadowing values of the points on the object is determined on page ("Real-time, realistic global illumination... it requires integration over the hemisphere of lighting directions at each point (light integration), and it must account for bouncing/occlusion effects, like shadows, due to intervening matter along light paths from sources to receivers (light transport complexity).") and on page 5 second column first paragraph lines ("We tag each direction sd with an occlusion bit, 1 ( ) p dV s - , indicating whether sd is in the hemisphere and intersects O again (i.e., is self-shadowed by O)."), therefore the depth of the sampled point is tested to determine visibility of the current sampled point in the current direction. Sloan fails to teach a sampled point represented in the object positions texture. Burke teaches an object positions texture that represents the depth of the sampled points in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g.,

"IJK")"). The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 1.

Regarding claim 20, Sloan teaches determining cosine terms on page 1 second column second paragraph lines 2-8 ("...object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral."), where it described that the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing on page 2 first column third paragraph lines 1-3 ("Shadow maps, containing depths from the light source's point of view...") and in the caption of Figure 2 lines 4-6 ("...a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing..."), and determining and accumulating radiance transfer contributions over the sampled points in each direction in the caption of Figure 2 lines 1-10 ("A transfer vector at a particular point on the surface represents how the surface responds to incident light at that point...matrix transforms the lighting coefficients into the coefficients of a spherical function...The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point..."). However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the method illustrated in Figure 3 of the applicant's Specification, because reversal of the loops would optimize the radiance processing when executed on graphics hardware, as taught by Purcell in section 3.2 4th paragraph lines 1-5 – 10-14 ("...we present an optimization to minimize the total number of passes...There

are various strategies for nesting these loops...The following is a more efficient algorithm...“), where the nested loops are reversed to enable efficient execution on a graphics processor, as shown in the pseudo-code in section 3.2 in the right column. The motivation to combine the teachings of Sloan, Burke and Purcell is equivalent to the motivation of claim 3.

Claims 2, 5, 12, 13 and 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan, in view of Morioka et al.(hereinafter “Morioka”, US Patent 6,333,742) in further view of Burke and in further view of Purcell.

Regarding claim 2, Sloan teaches hardware-accelerated processing, which is implied to be performed on a computer system, of a radiance transfer coefficients computation for a set of points sampled over a modeled object, on page1 second column third paragraph lines 1-7 (“The resulting transfer functions are represented as a dense set of vectors or matrices over its surface...The graphics hardware can dynamically sample incident radiance at a number of points.“), on page 8 first column fourth paragraph lines 8-10 (“Using graphics hardware, incident lighting can be sampled every frame and at multiple points near the object allowing dynamic, local lighting.“) and on page 7 section 8 second paragraph lines 4-7 and second column lines 1-3 (“Our current implementation precomputes the transfer matrix p M at each point...we perform the matrix transform from equation (9) in software at each point...The result is a volume texture containing coefficients of transferred radiance...“), for use in rendering images of the object, as illustrated in Figure 1. Sloan also teaches calculating radiance transfer using software on page 6 first column section 6 second paragraph lines 1-3 (“Step 1 can load a precomputed environment map, evaluate analytic lighting models in software, or sample radiance using graphics

hardware."), therefore a radiance transfer processing program is executed on a computer system. Sloan teaches computing radiance transfer functions for each set of directions sampled about the object on page 1 second column second paragraph lines 1-8 (" ...we have a convex, diffuse object lit by an infinitely distant environment map. The object's "response" to its environment can be viewed as a *transfer function*, mapping incoming and outgoing radiance...A more complex integral captures how a concave object shadows itself, where the integrand is multiplied by an additional transport factor representing visibility along each direction."). Sloan also teaches rendering the object from the direction to produce a shadow buffer representing depth from the object in the direction for the set of points on page 2 first column third paragraph lines 1-3 ("Shadow maps, containing depths from the light source's point of view...") and in the caption of Figure 2 lines 4-6 ("...a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing..."). Sloan also teaches determining radiance transfer contribution of the set of sampled points for the currently iterated direction based on the determined cosine terms and shadowing on page 1 second column second paragraph lines 2-8 ("object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral. A more complex integral captures how a concave object shadows itself, where the integrand is multiplied by an additional transport factor representing visibility along each direction."), where it described that the radiance transfer is computed for all directions in response to a integral comprising the cosine terms and shadowing effects. Sloan teaches accumulating the radiance transfer contributions of the set of sampled points for the currently iterated direction with that that of previously iterated direction

on page5 second column second paragraph lines 1-7 ("For diffuse surfaces, at each point pεO we further compute the transfer vector by SH-projecting ...SH-projection to compute the transfers is performed by numerical integration over the direction samples s_d, summing into an accumulated transfer..."), where it is described that for each point on the surface the radiance transfer is accumulated for all directions and therefore computes the radiance transfer for the current as well as any previous direction. Sloan illustrates a rendered image of an object in a lighting environment based on accumulated radiance transfer contribution that is presented in the right of Figure 1. Sloan fails to teach a memory for storing program code of at least on pixel shader, a central processing unit to execute the radiance transfer coefficients processing program and a graphics processing unit programmable by and operating to execute the at least one pixel shader and an object positions and normals texture. Morioka teaches a memory for storing program code of at least one pixel shader and a radiance transfer coefficients processing program in column 17 lines 39-47 where it is described that program code, as illustrated in Figure 21, that is known in the art to be stored on a computer readable medium, performs pixel shading in step 7 and processes radiance in step 1 where it is described that the light intensity values are determined. Morioka also teaches a central processing unit operating to execute the radiance transfer coefficients processing program in column 16 lines 10-16 where it is described that the light source information, which includes the radiance value of the pixel, is process by the CPU as illustrated in Figure 16 as element 1. Morioka teaches a graphics processing unit in column 7 lines 24-33 where it is described that a geometry processor 2 that is illustrated in Figure 6, performs graphics processing and executes at least one pixel shader by a rendering processor as described in column 7 lines 34-38 and is illustrated as element 32 in Figure 6. Morioka teaches

the at least one pixel shader executing on a graphics processing unit performing texture operation from each direction sample about the object in column 12 lines 42-51 where it is described that the texture generator, which comprises the rendering processor that performs the same functionality of the disclosed graphics processing unit, performs texture operations for each pixel. The generated data is then sent to the shading circuit, which performs shading, and light intensity operations for each of a set of direction about the object as described in column 9 lines 1-4. Sloan and Morioka fail to teach an object positions and normals texture. Burke teaches creating an object positions texture representing positions of a set of points sample over the object mapped into a texture space and object normals texture representing normals of the set of sampled points mapped into the texture space in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")"). However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the method illustrated in Figure 3 of the applicant's Specification, because reversal of the loops would optimize the radiance processing when executed on graphics hardware, as taught by Purcell in section 3.2 4th paragraph lines 1-5 – 10-14 ("...we present an optimization to minimize the total number of passes...There are various strategies for nesting these loops...The following is a more efficient algorithm..."), where the nested loops are reversed to enable efficient execution on a graphics processor, as shown in the pseudo-code in section 3.2 in the right column. Therefore, it would have been obvious to one of ordinary skill in the art at the

time of invention to combine the teachings of Sloan, Morioka, Burke and Purcell because this combination would provide a radiance transfer process that iterates over a set of directions in the outer loop and iterates over sample points in the inner loop, thereby efficiently computing realistic self-shadowing and lighting effects through optimization of the process for improved functionality on graphics hardware.

Regarding claim 5, Sloan teaches determining cosine terms on page 1 second column second paragraph lines 2-8 ("...object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral."), where it described that the radiance transfer is computed for all directions in response to a integral comprising computed cosine terms applied to an integral, determining shadowing on page 2 first column third paragraph lines 1-3 ("Shadow maps, containing depths from the light source's point of view...") and in the caption of Figure 2 lines 4-6 ("...a particular point on the surface represents how the surface responds to incident light at that point, including global transport effects like self-shadowing..."), and determining and accumulating radiance transfer contributions over the sampled points in each direction in the caption of Figure 2 lines 1-10 ("A transfer vector at a particular point on the surface represents how the surface responds to incident light at that point...matrix transforms the lighting coefficients into the coefficients of a spherical function...The result is convolved with the model's BRDF kernel and evaluated at the view-dependent reflection direction to yield the result at one point..."). Therefore, as stated in the Specification on page 8 lines 24-28 ("As compared to the previous PRT preprocess pseudo-code 300 of Figure 3, the order of the inner and outer loops of the hardware-accelerated PRT preprocess 400 are reversed to be more suitable for GPU

execution."), the code illustrated in Figure 4 is a reversed representation of the code of Figure 3, in which when it is reversed, produces the same results of the prior art as stated on page 12 lines 2-4. However, though Sloan teaches iteratively calculating for each point, the radiance in a plurality of directions, it would have been obvious to one of ordinary skill in the art at the time of invention to reverse the outer and inner loops of the method illustrated in Figure 3 of the applicant's Specification, because reversal of the loops would optimize the radiance processing when executed on graphics hardware, as taught by Purcell in section 3.2 4th paragraph lines 1-5 – 10-14 ("...we present an optimization to minimize the total number of passes...There are various strategies for nesting these loops...The following is a more efficient algorithm..."), where the nested loops are reversed to enable efficient execution on a graphics processor, as shown in the pseudo-code in section 3.2 in the right column. The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 13, Sloan fails to teach the limitations. Burke teaches tht the object positions texture contains an arrangement of data values representing the position of each of the sampled points mapped into the texture space in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")"). The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 15, Sloan fails to teach the limitations. Burke teaches object normals texture contains an arrangement of data values representing the surface normal at each of the sampled points mapped into the texture space in paragraph 0035 lines 4-10 ("For each sampled

point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ")...and normal information (e.g., "IJK")"). The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 16, Sloan teaches that the set of directions are to uniformly distributed points on a unit sphere on page 2 first column first paragraph lines 12-16 ("...the coefficients of a spherical function representing self-scattered incident radiance at each point. This function is convolved with the object's BRDF and then evaluated at the view-dependent reflection direction...") and on page 4 second column seventh paragraph lines 1-3 ("...transfer the incident radiance Lp(s) into a whole sphere of transferred radiance..."), where it is described that the set of points are distributed uniformly on the surface of the object, therefore the set of directions would also be computed uniformly on a unit sphere, as shown in Figure 2, because the set of directions are computed for all the points on the object as described on page 5 second column third paragraph lines 1-3 ("The vector Mp or matrix p M at each point p is initialized to 0 before the shadow pass, which then sums over all s_d at every p."), and as illustrated in the Figure located in the second column of page 5.

Regarding claim 17, Sloan teaches rendering the object from the direction comprises the object as an orthographic camera projection whose view direction is set to the current direction on page 2 first column first paragraph lines 14-16 ("...evaluated at the view-dependent reflection direction to produce the final shading."), where it is described that the shading is perform based on the view therefore the object is rendered based on the particular view direction, as described on page 6 first column section 6 first paragraph line 3 and step 4 ("Rendering O requires the

following steps at run-time:... the radiance vector resulting from step 3 is convolved with O's BRDF at p, and then evaluated at the view-dependent reflection direction R").

Regarding claim 18, Sloan teaches that the occlusion and shadowing values of the points on the object is determined on page ("Real-time, realistic global illumination... it requires integration over the hemisphere of lighting directions at each point (light integration), and it must account for bouncing/occlusion effects, like shadows, due to intervening matter along light paths from sources to receivers (light transport complexity).") and on page 5 second column first paragraph lines ("We tag each direction sd with an occlusion bit, 1 ( ) p dV s - , indicating whether sd is in the hemisphere and intersects O again (i.e., is self-shadowed by O)."), therefore the depth of the sampled point is tested to determine visibility of the current sampled point in the current direction. Sloan fails to teach a sampled point represented in the object positions texture. Burke teaches an object positions texture that represents the depth of the sampled points in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects) (e.g., "RGB"), and normal information (e.g., "IJK")"). The motivation to combine the teachings of Sloan, Morioka, Burke, and Purcell is equivalent to the motivation of claim 2.

Regarding claim 19, Sloan teaches code or software executable on the graphics accelerating hardware of the computer on page1 second column third paragraph lines 1-7 ("The resulting transfer functions are represented as a dense set of vectors or matrices over its surface...The graphics hardware can dynamically sample incident radiance at a number of points.") and on page 7 section 8 second paragraph lines 4-7 and second column lines 1-3 ("Our

current implementation precomputes the transfer matrix p M at each point…we perform the matrix transform from equation (9) in software at each point…The result is a volume texture containing coefficients of transferred radiance…"), to perform texture –based operations is a pixel shader, as described on page 7 first column fifth paragraph lines 4-7 and second column first paragraph lines 1-5("At run-time, we perform the matrix transform from equation (9) in software at each point in the volume and upload the result to the graphics hardware. The result is a volume texture containing coefficients of transferred radiance (L'_p) which is applied to R. Then in a pixel shader this transferred radiance is used to light the receiver."), executable on a programmable graphics processing unit, such as the programmable graphics processor described on page 6 first column section 6 third paragraph line 6 ("DirectX 8.1 pixel shaders").

Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan in view of Burke in further view of Purcell, and in further view of Arvo et al.(hereinafter "Avro", "*Monte Carlo Ray Tracing*").

Regarding claim 9, Sloan, Burke and Purcell fail to teach the limitations. Arvo teaches a set of directions that are generated as uniformly distributed points on a unit sphere based on a mapping from the unit square to the sphere on page 41 first paragraph lines 1-5 ("…uniformly distributed samples in the unit square are mapped to uniformly distributed samples in the range. Such mappings also preserve *stratification*, also known as *jitter sampling*…") and on page 23 fifth paragraph lines 1-4 ("…a set of jittered points on the unit square can be easily transformed to a set of jittered points on the hemisphere…"), where it is described that the points form the unit square are mapped on to the hemisphere or unit sphere, as described on page 23 second

paragraph lines 1-3 ("To choose reflected ray directions for zonal calculations or distributed ray tracing, we can think of the problem as choosing points on the unit sphere or hemisphere…"). It would have been obvious to one of ordinary skill in the art to combine the teachings of Sloan, Burke, Purcell and Arvo because this combination would provide a smooth representation of the lighting of a surface through the use of jittered sampling, which prevents unwanted aliasing effects.

Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan, in view of Morioka, in further view of Burke, in further view of Purcell, and in further view of Airey et al. (hereinafter "Airey", US Patent 6,650,327).

Regarding claim 14, Sloan fails to tech the limitations. Burke teaches an object positions texture in paragraph 0035 lines 4-10 ("For each sampled point, the model data receiving unit 110 receives data representing the position of the point (e.g., "XYZ") the color of the point (including vertex color, mapped textures, and static lighting effects)…"). However, Burke, Purcell and Morioka fail to teach storing texture in a floating point number format. Airey teaches storing texture in floating point number format in column 4 lines 18-20 ("Texturing, fog, and antialiasing all operate on floating point numbers. The texture map stores floating point texel values."). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Sloan, Morioka, Burke, Purcell and Airey because this combination would provide an efficient storage of textures in floating point number format that enables a more accurate processing to take place on the graphics hardware.

Response to Arguments

Applicant's arguments with respect to claims 1-20 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Said Broome whose telephone number is (571)272-2931. The examiner can normally be reached on 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on (571)272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

S. Broome
3/3/07 SB

UChauhan
ULKA CHAUHAN
SUPERVISORY PATENT EXAMINER